



## **Module 5: Hands-On: Creating Ansible Playbooks**

## Installing software using Ansible Playbook:

Prerequisite:

1. 3 AWS Servers
2. Keyless Access configured between master and slaves
3. Configuration of slaves in Ansible Master's /etc/ansible/hosts
4. Slaves are called slave1 and slave2 respectively

## Creating Ansible Playbook:

**Step 1:** Edit an Ansible Playbook in a new folder called 'ansible', by using the following commands:

```
$ mkdir ansible
$ cd ansible
$ sudo nano play.yaml
```

**Step 2:** Edit the play.yaml with the following syntax:

```
---
- hosts: slave1
  become: yes
  name: Installing apache2 on slave1
  tasks:
    - name: Install apache2
      apt: name=apache2 state=latest
- hosts: slave2
  become: yes
  name: Installing nginx on slave2
  tasks:
    - name: installing nginx
      apt: name=nginx state=latest
```

```
GNU nano 2.9.3 play.yaml
--
- hosts: slave1
  become: yes
  name: Installing nginx on slave1
  tasks:
    - name: Installing
      apt: name=nginx state=latest

- hosts: slave2
  become: yes
  name: Installing apache2 on slave2
  tasks:
    - name: Installing
      apt: name=apache2 state=latest
```

**Step 3:** Run the Ansible Playbook using the following command:

```
$ ansible-playbook play.yaml
```

```
ubuntu@ip-172-31-24-91:~/ansible$ ansible-playbook play.yaml
PLAY [Installing nginx on slave1] *****
TASK [Gathering Facts] *****
ok: [slave1]

TASK [Installing] *****
changed: [slave1]

PLAY [Installing apache2 on slave2] *****
TASK [Gathering Facts] *****
ok: [slave2]

TASK [Installing] *****
changed: [slave2]

PLAY RECAP *****
slave1      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
slave2      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

You have successfully installed NGINX and Apache2 on slave machines, based on their name.

## Executing scripts using Ansible Playbook:

**Step 1:** Create a sample script in the Ansible folder. Here we are creating a script to enter some text in a file

```
$ sudo nano script.sh
```

Write the following syntax in the script:

```
#!/bin/sh  
echo "hello world" > /var/www/html/1.html
```

**Step 2:** Let's create the Playbook now which is going to execute this script. Let's add a task to our previously written Playbook for slave2.

```
---  
- hosts: slave1  
  become: yes  
  name: Installing apache2 on slave1  
  tasks:  
    - name: Install apache2  
      apt: name=apache2 state=latest  
- hosts: slave2  
  become: yes  
  name: Installing nginx on slave2  
  tasks:  
    - name: installing nginx  
      apt: name=nginx state=latest  
    - name: Running a script  
      script: script.sh
```

**Step 3:** Let's run the script now using the following syntax:

```
$ ansible-playbook play.yaml
```

```
ubuntu@ip-172-31-24-91:~/ansible$ ansible-playbook play.yaml
PLAY [Installing nginx on slave1] *****
TASK [Gathering Facts] *****
ok: [slave1]
TASK [Installing] *****
ok: [slave1]
PLAY [Installing apache2 on slave2] *****
TASK [Gathering Facts] *****
ok: [slave2]
TASK [Installing] *****
ok: [slave2]
TASK [run script] *****
changed: [slave2]
PLAY RECAP *****
slave1      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
slave2      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Let's verify the output by going to our browser, and navigating to /1.html



hello world

With this, we have successfully completed our Ansible Playbook hands-on.